Semigroups and sequential importance sampling for multiway tables

Ruriko Yoshida
Department of Statistics, University of Kentucky
and
Jing Xi
Department of Statistics, University of Kentucky
and
Shaoceng Wei
Department of Statistics, University of Kentucky
end
Feng Zhou
Department of Statistics, University of Kentucky
end
David Haws
Department of Statistics, University of Kentucky

November 29, 2011

Abstract

When an interval of integers between the lower bound l_i and the upper bound u_i is the support of the marginal distribution $n_i|(n_{i-1},\ldots,n_1)$, Chen et al. (2005a) noticed that sampling from the interval at each step, for n_i during a sequential importance sampling (SIS) procedure, always produces a table which satisfies the marginal constraints. However, in general, the interval may not be equal to the support of the marginal distribution. In this case, the SIS procedure may produce tables which do not satisfy the marginal constraints, leading to rejection (Chen et al., 2006). In this paper we consider the uniform distribution as the target distribution. First we show that if we fix the number of rows and columns of the design matrix of the model for contingency tables then there exists a polynomial time algorithm in terms of the input size to sample a table from the set of all tables satisfying all marginals defined by the given model via the SIS procedure without rejection. We then show experimentally that in general the SIS procedure may have large rejection rates even with small tables. Further we show that in general the classical SIS procedure in (Chen et al.,

2005a) can have a large rejection rate whose limit is one. When estimating the number of tables in our simulation study, we used the univariate and bivariate logistic regression models since under this model the SIS procedure seems to have higher rate of rejections even with small tables.

Keywords: Contingency tables, estimating number of tables, generating functions, semigroups, sequential importance sampling.

1 Introduction

Sampling from two-way and multiway contingency tables has a wide range of applications such as computing exact p-values of goodness-of-fit, estimating the number of contingency tables satisfying given marginal sums and more (Besag & Clifford, 1989; Chen et al., 2005a; Diaconis & Efron, 1985; Guo & Thompson, 1992). For some problems, such as sparse tables, the data of interest does not permit the use of asymptotic methods. In such cases, one can apply Monte Carlo Markov Chain (MCMC) procedures using *Markov bases* (Diaconis & Sturmfels, 1998). In order to run MCMC over the state space, all states must be connected via a Markov chain. A Markov basis is a set of moves on all contingency tables (the state space) guaranteed to be connected via a Markov chain (Diaconis & Sturmfels, 1998). One important quality of a Markov basis is that the moves will work for any marginal sums under a fixed model. The two major advantages to using a MCMC approach, if a Markov basis is already known, is that it is easy to program, and it is not memory intensive. However MCMC methods are not without drawbacks where one bottleneck is the computation of a Markov basis. In fact, for 3-way contingency tables with fixed 2-margins, De Loera & Onn (2005) showed that the number of Markov basis elements can be arbitrary. To try to circumvent the difficulty of computing a Markov basis which may be large, Chen et al. (2005b); Bunea & Besag (2000) studied computing a smaller set of moves by allowing entries of the contingency table to be negative. The trade off to this approach is longer running time of the Markov chains. Even using a standard MCMC approach, to sample a table independently from the distribution, the Markov chains can take a long time to converge to a stationary distribution in order to satisfy the independent assumption. Lastly, it is not clear in general how long the chain must be run to converge.

A sequential importance sampling (SIS) procedure is easy to implement and was first applied to sampling two-way contingency tables under the independence model in (Chen et al., 2005a). It

proceeds by simply sampling cell entries of the contingency table sequentially such that the final distribution approximates the target distribution. This method will terminate at the last cell and sample independently and identically distributed (iid) tables from the proposal distribution. Thus the SIS procedure does not require expensive or prohibitive pre-computations, as is the case of computing a Markov basis for a MCMC approach. Second, when attempting to sample a single table, the SIS procedure is guaranteed to sample a table from the distribution, where in an MCMC approach the chain may require a long time to run in order to satisfy the independent condition. In these regards, the SIS overcomes the disadvantages of MCMC but presents a new set of problems. One major difficulty is computing the marginal distribution of each cell. Typically an interval from which the support of the marginal distribution lies is computed using Integer Programming (IP), Linear Programming (LP), or the Shuttle Algorithm (Dobra & Fienberg, 2010). When the support of the marginal distribution does not equal the interval, the SIS procedure may reject the partially sampled table. The SIS has been successful on two-way contingency tables due to the fact that the computed interval often equals the support of the marginal distribution. For example, under the independence model, the SIS procedure will always produce tables satisfying the marginal sums, i.e. there are no rejections (Chen et al., 2006). Moreover, for zero-one two-way contingency tables (Chen et al., 2005a) provided an algorithm to sample using the SIS with Conditional Poisson distributions which also avoids rejections and relies on the Gale-Ryser Theorem. Chen et al. (2006) extended the SIS to multiway contingency tables, and gave excellent algebraic interpretations of precisely when an interval will equal the support of the marginal distribution. Regardless, until now, one of the major disadvantages of the SIS is the fact that rejections lead to increased computational time.

In this paper we focus on the uniform distribution as the target distribution. Here we show that if we fix the number of rows and columns of the *design matrix* defined in Subsection 2.1, then there exists a polynomial time algorithm in terms of the input size to sample a table from the set of all tables satisfying all given marginals via the SIS procedure without rejection. For the proof we use the notion of the *semigroup* (defined in Subsection 2.3), of the rays of the design matrix and *short generating functions* of a set. Then we show that the classical SIS procedure in Chen et al. (2005a) can have a large rejection rate whose limit is one, i.e., it can be very close to one.

In order to assess the rejection rate we conduct a simulation study on the *univariate and bivariate logistic regression* models. In our simulation study we show that in general the SIS procedure can have a high rejection rate even with small tables.

Notation is covered in Subsection 2.1 and the SIS procedure is described in detail in Subsection 2.2. The semigroup of the design matrices is discussed in Subsection 2.3. In Section 3 we will show that if we fix the number of rows and columns of the design matrix of the given model for contingency tables then there exists a polynomial time algorithm in terms of input size to sample a table from the set of all tables satisfying the marginals defined by the given model. In Section 4 we show by an example that a classical SIS procedure can have a large rejection rate in general and then we show experimental results with the SIS in order to assess the rejection rate under the univariate/bivariate logistic regression models.

2 Preliminaries

2.1 Basic notation

Let $\mathbf{n}$ be a contingency table with k cells. In order to simplify the notation, we denote by $\mathcal{X} = \{1, \ldots, k\}$ the sample space of contingency tables.

Let $\mathbb{Z}_+$ be the set of nonnegative integers, i.e., $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$ and let $\mathbb{Z}$ be the set of all integers, i.e., $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. Without loss of generality, in this paper, we represent a table by a vector of counts $\mathbf{n} = (n_1, \ldots, n_k)$. With this point of view, a contingency table $\mathbf{n}$ can be regarded as a function $\mathbf{n} : \mathcal{X} \longrightarrow \mathbb{Z}_+$, and it can also be viewed as a vector $\mathbf{n} \in \mathbb{Z}_+^k$.

The fiber of an observed table $\mathbf{n}_{\text{obs}}$ with respect to a function $T : \mathbb{Z}_+^k \longrightarrow \mathbb{Z}_+^d$ is the set

$$\mathcal{F}_T(\mathbf{n}_{\text{obs}}) = \left\{ \mathbf{n} \;\mid\; \mathbf{n} \in \mathbb{Z}_+^k \;,\; T(\mathbf{n}) = T(\mathbf{n}_{\text{obs}}) \right\} . \tag{1}$$

When the dependence on the specific observed table is irrelevant, we will write simply $\mathcal{F}_T$ instead of $\mathcal{F}_T(\mathbf{n}_{\text{obs}})$.

In a mathematical statistics framework, the function T is usually the minimal sufficient statistic of some statistical model and the usefulness of enumeration of the fiber $\mathcal{F}_T(\mathbf{n}_{\text{obs}})$ follows from classical theorems such as the Rao-Blackwell theorem, see e.g. (Shao, 1998).

When the function T is linear, it can be extended in a natural way to a homomorphism from

$\mathbb{R}^k$ to $\mathbb{R}^d$. The function T is represented by an $d \times k$-matrix A_T, and its element $A_T(\ell, h)$ is

$$A_T(\ell, h) = T_\ell(h), \tag{2}$$

where T_ℓ is the ℓ-th component of the function T. The matrix A_T is called the *design matrix* of the model T. In terms of the matrix A_T, the fiber $\mathcal{F}_T$ can be easily rewritten in the form:

$$\mathcal{F}_T = \left\{ \mathbf{n} \mid \mathbf{n} \in \mathbb{Z}_+^k \ , \ A_T \mathbf{n} = A_T \mathbf{n}_{\text{obs}} \right\} . \tag{3}$$

When the context is clear, we will simply write A instead of A_T.

Remark 2.1. *Since the cell counts of a contingency table are nonnegative integers and usually the sufficient statistics are a set of marginals of cell counts, the design matrix A of a model T is a nonnegative $d \times k$ matrix.*

2.2 Sequential importance sampling

Let $\mathcal{F}_T$ be the set of all tables satisfying marginal conditions (for example, under the independence model, all tables satisfying given row and column sums). Let $p(\mathbf{n})$, for any $\mathbf{n} \in \mathcal{F}_T$, be the uniform distribution over $\mathcal{F}_T$, so $p(\mathbf{n}) = 1/|\mathcal{F}_T|$. Let $q(\cdot)$ be a trial distribution such that $q(\mathbf{n}) > 0$ for all $\mathbf{n} \in \mathcal{F}_T$. Then we have

$$\mathbb{E}\left[\frac{1}{q(\mathbf{n})}\right] = \sum_{\mathbf{n} \in \mathcal{F}_T} \frac{1}{q(\mathbf{n})} q(\mathbf{n}) = |\mathcal{F}_T|.$$

Thus we can estimate $|\mathcal{F}_T|$ by

$$\widehat{|\mathcal{F}_T|} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{q(\mathbf{n_i})},$$

where $\mathbf{n_1}, \ldots, \mathbf{n_N}$ are tables drawn iid from $q(\mathbf{n})$. Here, this proposed distribution $q(\mathbf{n})$ is the distribution (approximate) of tables sampled via the SIS.

If we vectorize the table $\mathbf{n} = (n_1, \cdots, n_k)$ then by the multiplication rule we have

$$q\Big(\mathbf{n} = (n_1, \cdots, n_k)\Big) = q(n_1) q(n_2|n_1) q(n_3|n_2, n_1) \cdots q(n_k|n_{k-1}, \ldots, n_1).$$

Since we sample each cell count of a table from a interval we can easily compute $q(n_i|n_{i-1}, \ldots, n_1)$ for $i = 2, 3, \ldots, k$.

When an interval of integers between the lower bound l_i and the upper bound u_i is the support of the marginal distribution $n_i|(n_{i1}, \ldots, n_1)$ for n_i, Chen et al. (2006) noticed that one can sample

a value from the interval at each step for n_i from the interval $[l_i, u_i]$ and this procedure always produces a table which satisfies the marginal constraints. Therefore if we can obtain l_i and u_i for each n_i sequentially we can apply the SIS. Usually we obtain l_i and u_i for each n_i by Integer Programming (IP) to obtain tight bounds, namely we solve the linear integer programming problem for the lower bound:

$$\begin{aligned} &\min x_i \\ &\text{s.t. } \mathbf{a}_i x_i + \cdots + \mathbf{a}_k x_k = b - \left(\mathbf{a}_1 x_1^* + \cdots + \mathbf{a}_{i-1} x_{i-1}^*\right), \\ &\quad x_i, \ldots x_k \in \mathbb{Z}_+, \end{aligned} \tag{4}$$

where $x_1^*, \ldots x_{i-1}^*$ are integers already sampled by the SIS, $\mathbf{b} = A\mathbf{n}_{\text{obs}}$, and $\mathbf{a}_j$ is the jth column of A. To compute the upper bound via IP we set max instead of min in Equation (4). One can approximate these bounds by linear programming (LP) or the Shuttle Algorithm (Buzzigoli & Giusti, 1998), however they might not give tight bounds.

Based on this observation Chen et al. (2006) developed a sequential importance sampling (SIS) method to sample a table from $\mathcal{F}_T$. The outline of the SIS procedure is the following:

Algorithm 2.2. *[Sequential importance sampling procedure]*

1. *For $i = 1, \ldots, k$ do:*
   (a) *Compute l_i and u_i by solving an integer programming problem* (4).
   (b) *Sample an integer x_i^* from the interval $[l_i, u_i]$ according to the distribution q.*
2. *Return the table $\mathbf{x}^* = (x_1^*, \ldots, x_k^*)$.*

Remark 2.3. *If we want to estimate $|\mathcal{F}_T|$ then q is the uniform distribution over $[l_i, u_i] \cap \mathbb{Z}$. Thus we sample x_i^* from $[l_i, u_i] \cap \mathbb{Z}$ with a probability $1/(u_i - l_i + 1)$.*

When we have rejections, this means that we are sampling tables from a bigger set $\mathcal{F}_T^*$ such that $\mathcal{F}_T \subset \mathcal{F}_T^*$. In this case, as long as the conditional probability $q(n_i | n_{i-1}, \ldots, n_1)$ for $i = 2, 3, \ldots$ and $q(n_1)$ are normalized, $q(\mathbf{n})$ is normalized over $\mathcal{F}_T^*$ since

$$\begin{aligned} \textstyle\sum_{\mathbf{n} \in \mathcal{F}_T^*} q(\mathbf{n}) &= \textstyle\sum_{n_1, \ldots n_k} q(n_1) q(n_2|n_1) q(n_3|n_2, n_1) \cdots q(n_k | n_{k-1}, \ldots, n_1) \\ &= \textstyle\sum_{n_1} q(n_1) \left[\sum_{n_2} q(n_1|n_2) \left[\cdots \left[\sum_{n_k} q(n_k | n_{k-1}, \ldots, n_1) \right] \right] \right] \\ &= 1. \end{aligned}$$

Thus we have

$$\mathbb{E}\left[\frac{\mathbb{I}_{\mathbf{n}\in\mathcal{F}_T}}{q(\mathbf{n})}\right] = \sum_{\mathbf{n}\in\mathcal{F}_T^*} \frac{\mathbb{I}_{\mathbf{n}\in\mathcal{F}_T}}{q(\mathbf{n})} q(\mathbf{n}) = |\mathcal{F}_T|,$$

where $\mathbb{I}_{\mathbf{n}\in\mathcal{F}_T}$ is an indicator function for the set $\mathcal{F}_T$. By the law of large numbers this estimator is unbiased.

2.3 Semigroup

Consider the following system of linear equations and inequalities:

$$A\mathbf{x} = \mathbf{b}, \;\; \mathbf{x} \geq 0, \tag{5}$$

where $A \in \mathbb{Z}^{d\times k}$ and $\mathbf{b} \in \mathbb{Z}^d$. Suppose the solution set $\{x \in \mathbb{R}^k : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\} \neq \emptyset$.

Note that there exists an integral solution for the system in (5) if and only if $\mathbf{b}$ is in the *semigroup* generated by the column vectors $\mathbf{a}_1, \ldots, \mathbf{a}_k$ of A, that is, the set of all nonnegative integer combinations of the columns of A, namely

$$Q = Q(A) = \{\mathbf{a}_1 x_1 + \cdots + \mathbf{a}_k x_k \mid x_1, \ldots, x_k \in \mathbb{Z}_+\}. \tag{6}$$

Let $K = K(A)$ be the cone generated by the columns $\mathbf{a}_1, \ldots, \mathbf{a}_k$ of A, that is:

$$K = K(A) = \{\mathbf{a}_1 x_1 + \cdots + \mathbf{a}_k x_k \mid x_1, \ldots, x_k \in \mathbb{R}_+\}.$$

The lattice $L = L(A)$ generated by the columns $\mathbf{a}_1, \ldots, \mathbf{a}_k$ of A is:

$$L = L(A) = \{\mathbf{a}_1 x_1 + \cdots + \mathbf{a}_n x_n \mid x_1, \ldots, x_k \in \mathbb{Z}\}.$$

The semigroup $Q_{sat} = K \cap L$ is called the *saturation* of the semigroup Q. It follows that $Q \subset Q_{sat}$ and we call Q *saturated* if $Q = Q_{sat}$ (this also is called *normal*). We define $H = Q_{sat} \setminus Q$ the set of *holes* of the semigroup Q.

Remark 2.4. *If $b \in H \subset Q_{sat}$, then the system*

$$Ax = b, x \geq 0, x \in \mathbb{R}^k$$

has a feasible solution. However, the system

$$Ax = b, x \geq 0, x \in \mathbb{Z}^k$$

does not have a feasible solution.

3 Time complexity of the SIS procedure without rejection

In 2002, Barvinok & Woods (2003) introduced an algorithm to encode all integral vectors $b \in \mathbb{Z}^k$ in a semigroup $Q(A)$ as a *short rational generating function* in polynomial time in terms of input size when d and k are fixed (Lemma 3.2 stated below).

One might ask the time complexity of the SIS procedure without rejection in general. Using the results from (Barvinok, 1994; Barvinok & Pommersheim, 1999; Barvinok & Woods, 2003), we can prove that the SIS procedure without rejection can be solved in polynomial time in fixed d and k (Theorem 3.1). In order to prove the theorem, we will use the *multivariate generating function* of a set $X \subset \mathbb{Z}^d$, $f(X;x)$. Namely, if $X \subset \mathbb{Z}^d$, we define the generating function

$$f(X;x) = \sum_{s \in X} x^s,$$

where x^s denotes $x_1^{s_1} \cdots x_d^{s_d}$ with $s = (s_1, \ldots, s_d)$. If $X = P \cap \mathbb{Z}^d$ with fixed d, where P is a rational convex polyhedron, or if $X = Q$ with fixed d and k, then Barvinok (1994) and Barvinok & Woods (2003), respectively, showed that $f(X;x)$ can be written in the form of a polynomial-size sum of rational functions of the form:

$$f(X;x) = \sum_{i \in I} \gamma_i \frac{x^{\alpha_i}}{\prod_{j=1}^{d}(1 - x^{\beta_{ij}})}. \tag{7}$$

Herein, I is a finite (polynomial size) index set and all the appearing data $\gamma_i \in \mathbb{Q}$ and $\alpha_i, \beta_{ij} \in \mathbb{Z}^d$ is of size polynomial. If a rational generating function $f(X;x)$ is polynomial size in the total bit size of inputs, then $f(X;x)$ is called a *short rational generating function*. As an example, if P is the one-dimensional polytope $[0, N]$, $N \in \mathbb{Z}_+$, then $f(P \cap \mathbb{Z}; x) = 1 + x + x^2 + \cdots + x^N$, $f(P \cap \mathbb{Z}; x)$ can be represented by a short rational generating function $(1 - x^{N+1})/(1 - x)$.

Theorem 3.1. *Suppose we fix d and k. Assume that the design matrix for the model is $A \geq 0$. There is a polynomial time algorithm in terms of the input size to sample a table* $\mathbf{n}$ *via the SIS procedure without rejection from the set $|\mathcal{F}_T|$ where $\mathcal{F}_T$ is defined in* (3).

Before proofs of Theorem 3.1, we would like to state lemmas from (Barvinok & Woods, 2003) and (Barvinok & Pommersheim, 1999).

Lemma 3.2 ((7.3) in (Barvinok & Woods, 2003))**.** *Suppose we fix d and k. Let $Q = Q(A)$. Then there exists an integer $s = s(d)$ and the generating function $f(Q;x)$ for the semigroup Q can be*

computed in polynomial time in terms of the input size as a short rational generating function in the form of

$$f(S;x) = \sum_{i \in I} \gamma_i \frac{x^{u_i}}{(1-x^{v_{i1}}) \cdots (1-x^{v_{is}})},$$

where $\gamma_i \in \mathbb{Q}, u_i, v_{ij} \in \mathbb{Z}^d$ *and* $v_{ij} \neq 0$ *for all* i, j.

Lemma 3.3 (Theorem 4.4 in (Barvinok & Pommersheim, 1999))**.** *Suppose we fix* d *and suppose* $P \subset \mathbb{R}^d$ *is a rational convex polyhedron. Then the generating function* $f(P \cap \mathbb{Z}^d; x)$ *can be computed in polynomial time in terms of the input size as a short rational generating function in the form of* (7).

The following lemma is an extension of Theorem 3.6 in (Barvinok & Woods, 2003).

Lemma 3.4. *Suppose we fix* d. *Let* $S_1, S_2 \subset \mathbb{R}^d$ *such that* S_1 *is unbounded and* S_2 *is finite. Suppose there exists a vector* $l \in \mathbb{R}^d$ *such that* $\langle l, \mathbf{x} \rangle < 0$ *for all* $\mathbf{x} \in S_1$, *where* $\langle \cdot, \cdot \rangle$ *denotes a dot product of vectors, then there exists a polynomial time algorithm which given* $f_1(S_1; x)$ *and* $f_2(S_2; x)$, *the short generating function for* S_1 *and* S_2, *respectively, computes* $f(S; x)$ *for* $S = S_1 \cap S_2$ *in the form*

$$f(S;x) = \sum_{i \in I} \gamma_i \frac{x^{u_i}}{(1-x^{v_{i1}}) \cdots (1-x^{v_{is}})},$$

where $s \leq 2 \cdot d, \gamma_i \in \mathbb{Q}, u_i, v_{ij} \in \mathbb{Z}^d$ *and* $v_{ij} \neq 0$ *for all* i, j.

Proof. Since there exists a vector $l \in \mathbb{R}^d$ such that $\langle l, x \rangle < 0$ for all $\mathbf{x} \in S_1$, we can apply the proof for Lemma 3.4 in (Barvinok & Woods, 2003) to prove this lemma. □

Remark 3.5. *Similarly we can apply the same way to prove that if we fix* d *and* $S_1, S_2 \subset \mathbb{R}^d$ *are the two unbounded sets such that there exists a vector* $l \in \mathbb{R}^d$ *such that* $\langle l, \mathbf{x} \rangle < 0$ *for all* $\mathbf{x} \in S_1$ *and* $\langle l, \mathbf{x} \rangle < 0$ *for all* $\mathbf{x} \in S_2$, *then there exists a polynomial time algorithm which given* $f_1(S_1; x)$ *and* $f_2(S_2; x)$, *the short generating function for* S_1 *and* S_2, *respectively, computes* $f(S; x)$ *for* $S = S_1 \cap S_2$ *in the form*

$$f(S;x) = \sum_{i \in I} \gamma_i \frac{x^{u_i}}{(1-x^{v_{i1}}) \cdots (1-x^{v_{is}})},$$

where $s \leq 2 \cdot d, \gamma_i \in \mathbb{Q}, u_i, v_{ij} \in \mathbb{Z}^d$ *and* $v_{ij} \neq 0$ *for all* i, j.

Using the generating function we have the following algorithm to sample a table $\mathbf{n}$ from $\mathcal{F}_T$ via the SIS procedure without rejection in general.

Algorithm 3.6. Input *System*

$$Ax = b, x \in \mathbb{Z}_+^k, \tag{8}$$

where $A = (a_1, ..., a_k) \geq 0$ *is a* $d \times k$ *matrix and* $b = (b_1, ..., b_d) = A_T \mathbf{n}_{\text{obs}}$*, where* $\mathbf{n}_{\text{obs}}$ *is the observed table, is a* d *dimensional vector.*

Output $\mathbf{n} = (n_1, \ldots, n_k)$ *sampled via the SIS without rejection and its probability to be picked from* $\mathcal{F}_T$*.*

Algorithm

1. *Set* $p = 1$.
2. *For* $i = 1, \ldots, k - d$ *do*
   (a) *Set the system* $A'x = b'$, $x \in \mathbb{Z}_+^{(k-i+1)}$*, where* $b' = b - (\sum_{j=1}^{i-1} A_j x_j)$ *and* $A' = (a_i, ..., a_k)$.
   (b) *Compute the semigroup* Q' *generated by* $(a_{i+1}, ..., a_k)$ *via generating function and compute the generating function for* $P \cap \mathbb{Z}^d$ *where* $P = \{x \in \mathbb{R}^d : b' - xa_i \geq 0\}$.
   (c) *Since* $a \in \mathbb{Z}_+^d \setminus \{0\}$, $\forall a \in Q'$*, we choose* $l = (-1, -1, \cdots, -1)$ *to compute* $Q' \cap P$ *via generating functions.*
   (d) *Sample* n_i *uniformly from* $Q' \cap P$.
   (e) $p = p \cdot (1/\#|Q' \cap P|)$.
3. *Find* $n_{k-d+1}, \ldots, n_k$ *by solving the system:*

$$A_{final} x = b_{final}, x \in \mathbb{Z}_+^d,$$

   where $b_{final} = b - (\sum_{j=1}^{k-d} A_j x_j)$ *and* $A_{final} = (a_{k-d+1}, ..., a_k)$
4. *Return* $\mathbf{n}$ *and its probability to be picked* p.

Lemma 3.7. $Q' \cap P$ *in Step 2c is not empty.*

Proof. The right hand side $b = A_T \mathbf{n}_{\text{obs}}$ from the input is computed from the observed table $\mathbf{n}_{\text{obs}}$ so there exists at least one solution in the system (8). Thus $b \in Q(A)$. Thus $Q' \cap P \neq \emptyset$ for $i = 1$. For $i = 2, \ldots (k-d)$, since $b' \in Q(A')$ where $Q(A')$ is the semigroup generated by $a_i, \ldots, a_k$, there exists a nonnegative integral solution for the system in $A'x = b'$, $x \in \mathbb{Z}_+^{(k-i+1)}$. Thus $Q' \cap P \neq \emptyset$. □

Remark 3.8. *The system in Step 3 has a solution as well since $b' \in Q(A')$.*

Proof for Theorem 3.1. We fix d and k. For Step 2b we can compute the generating functions $f_1(Q';x)$ and $f_2(P;x)$ in polynomial time in terms of input size if we fix d and k by Lemmas 3.2 and 3.3. One can compute the short generating function for $Q' \cap P$ in Step 2c in polynomial time by Lemma 3.4. For Step 2d one can sample a point from $Q' \cap P$ in polynomial time via the method of sampling showed in (Pak, 2000) since $Q' \cap P \subset P$. One can compute $\#|Q' \cap P|$ in polynomial time by using the Taylor expansion of the limit of the short generating function with x around $(1, 1, \ldots, 1)$ similarly to how we count the number of lattice points in (Barvinok & Pommersheim, 1999). $Q' \cap P \neq \emptyset$ by Lemma 3.7 and Remark 3.8 so we do not reject the sampled table. □

Remark 3.9. *Note that implementing Algorithm 3.6 seems not to be practical since it seems not to be possible in practice at this moment to implement an algorithm in Theorem 3.2 to compute the generating function for a semigroup in polynomial time with fixed d and k even though we have a computational time result.*

4 Computational experiments

In general the classical SIS procedure in (Chen et al., 2005a) can have an arbitrary large rejection rate. For example, if we have the following system

$$x_1 + \alpha \cdot x_2 = \alpha + 1,\ x_1, x_2 \in \mathbb{Z}_+,$$

where $\alpha \in \mathbb{N}$ is an arbitrary large positive integer. Then by integer programming we have $l_1 = 0$ and $u_1 = \alpha + 1$. However, only two integral points in the interval $[0,\ \alpha + 1]$ for x_1 can give a solution to the system, namely $x_1 = 1$ or $x_1 = \alpha + 1$. Thus the rejection rate is $1 - \frac{2}{\alpha+2} = \frac{\alpha}{\alpha+2}$. Thus if we send $\alpha \to \infty$, the rejection rate becomes 1.

Thus it is interesting to assess the rejection rates in practice. In this section we conduct a simulation study under the univariate/bivariate logistic regression models. We chose this model since the SIS procedure seems to have a very high rejection rate even with small tables.

We used two types of random table generators:

1. A table generator using the Poisson distribution.

2. A table generator using the uniform distribution.

First we show how we generated random tables for Item 1.

- **Input**: A positive integer $k \geq 1$ and a positive rational number λ.
- **Output**: A randomly generated table $\mathbf{n} = (n_1, \ldots, n_k)$.

    1. For each cell n_i of the table X, pick a random number r uniformly from $(0, 1)$.
        - if $0 < r \leq \frac{5}{2^k}$, pick a random integer x uniformly from $(1, 1000)$ and assign $n_i = x$,
        - else if $\frac{5}{2^k} < r \leq \frac{1}{2}$, assign $n_i = 1 + Poisson(\lambda)$, where $Poisson(\lambda)$ is the Poisson distribution with a random variable λ,
        - else assign $n_i = 0$.
    2. If there is no cell in the table such that $n_i > 10$, then we randomly pick cell n_i uniformly. Pick a random integer x uniformly from $(1, 1000)$ and assign $n_i = x$.
    3. Return $\mathbf{n}$.

Now we show how we generated random tables for Item 2.

- **Input**: A positive integer $k \geq 1$.
- **Output**: A randomly generated table $\mathbf{n} = (n_1, \ldots, n_k)$.

    1. For each cell n_i of the table $\mathbf{n}$, pick a random number r uniformly from $(0, 1)$.
        - if $0 < r \leq \frac{5}{2^k}$, pick a random integer x uniformly from $(1, 1000)$ and assign $n_i = x$,
        - else if $\frac{5}{2^k} < r \leq \frac{1}{2}$, pick a random integer y uniformly from $(1, 10)$ and assign $n_i = y$,
        - else assign $n_i = 0$.
    2. If there is no cell in the table such that $n_i > 10$, then we randomly pick cell n_i uniformly. Pick a random integer x uniformly from $(1, 1000)$ and assign $n_i = x$.
    3. Return X.

In this section we consider the bivariate regression model since we verified that the semigroup is not normal in each experiment, i.e., there would be rejections with the SIS procedure. Further, the problems are small enough so that we can count the exact number of tables.

Let $\{1,\ldots,I\}$ and $\{1,\ldots,J\}$ be the set levels of two covariates. Let X_{1ij} and X_{2ij}, $i=1,\ldots,I$, $j=1,\ldots,J$, be the numbers of successes and failures, respectively, for level (i,j). The probability for success p_{1ij} is modeled as

$$\begin{aligned} \operatorname{logit}(p_{1ij}) = \log\left(\frac{p_{1ij}}{1-p_{1ij}}\right) = \mu + \alpha i + \beta j, \\ i = 1,\ldots,I, \quad j = 1,\ldots,J. \end{aligned} \tag{9}$$

The likelihood is written as

$$\begin{aligned} L(\alpha,\beta,\gamma) &\propto \prod_{i=1}^{I}\prod_{j=1}^{J}(1+\exp(\alpha+\beta i+\gamma j))^{-n_{+ij}} \\ &\quad \times \prod_{i=1}^{I}\prod_{j=1}^{J}\exp\left(\alpha n_{1ij}+\beta i n_{1ij}+\gamma j n_{1ij}\right) \\ &= \prod_{i=1}^{I}\prod_{j=1}^{J}(1+\exp(\alpha+\beta i+\gamma j))^{-n_{+ij}} \\ &\quad \times \exp\left(\alpha n_{1++}+\beta\sum_{i=1}^{I} i n_{1i+}+\gamma\sum_{j=1}^{J} j n_{1+j}\right). \end{aligned}$$

Thus the sufficient statistics for this model are X_{1++}, $\sum_{i=1}^{I} iX_{1i+}$, $\sum_{j=1}^{J} jX_{1+j}$, X_{+ij}, $\forall i,j$.

In our simulation study on estimating the number of tables via the SIS, we generated 100 tables under Model (9) for each experiment with sample size $N=100$.

One can find our software at `http://polytopes.net/code/SIS`.

5 Discussion and open problems

In this paper we showed that if we fix the number of rows and number of columns of the design matrix we can sample a table from the set of all tables given the marginals via the SIS procedure without rejection in polynomial time in terms of input size. Also we show that in general the rejection rate can be arbitrary large. However we have not been able to implement Algorithm 3.6 because at present it is very hard to implement a method in Theorem 3.2. It would be interesting to develop a polynomial time algorithm to sample a table from the set of all tables satisfying all marginal conditions in polynomial time with fixed dimensions.

option	model	I,J	time(sec)	reject
1	univariate	5	6.72	14
		6	8.84	14
		7	11.14	27
		8	11.56	27
		9	12.62	30
		10	14.78	30
	Bivariate	2,5	14.82	28
		2,6	18.32	30
		2,7	21.9	42
2	univariate	5	6.28	8
		6	7.4	14
		7	9.04	20
		8	19.8	28
		9	13.51	31
		10	15.38	27
	Bivariate	2,5	14.88	27
		2,6	18.33	30
		2,7	21.59	30

Table 1: In this experiment we set the sample size for each SIS procedure $N = 100$ and we generated 100 random tables. The first column represents the distribution of random table generator. For Item 1, we generated a random table according to the distribution with $\lambda = 1$. In this table the lower and upper bounds for each cell for sampling a table were computed by IP. The second column represents the model. Univariate means the univariate logistic regression model with label I. Bivariate means the bivariate logistic regression model with label I and J. The third column represents the level(s) of each variable. The fourth columns show the computational time in seconds. The last column shows the number of random tables out of 100 tables on which we had rejections.

6 Acknowledgements

The authors would like to thank Kevin Woods for discussion on the proof of Lemma 3.4.

References

Barvinok, A. (1994). Polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Math of Operations Research* 19 769–779.

Barvinok, A. & Pommersheim, J. (1999). An algorithmic theory of lattice points in polyhedra New perspectives in algebraic combinatorics, Berkeley, CA, 1996-1997. *Math. Sci. Res. Inst. Publ.* 38 91–147.

Barvinok, A. & Woods, K. (2003). Short rational generating functions for lattice point problems. *Journal of the American Mathematical Society* 16 957–979.

Besag, J. & Clifford, P. (1989). Generalized monte carlo significance tests. *Biometrika* 76 633–642.

Bunea, F. & Besag, J. (2000). Mcmc in $i \times j \times k$ contingency tables. In *Monte Carlo Methods (N. Madras, ed.)*, vol. 26. Fields Institute Communications, 25–36.

Buzzigoli, L. & Giusti, A. (1998). An algorithm to calculate the lower and upper bounds of the elements of an array given its marginals. *Statistical Data Protection (SDP'98) Proceedings* 131–147.

Chen, Y., Diaconis, P., Holmes, S. & Liu, J. (2005a). Sequential monte carlo methods for statistical analysis of tables. *American Statistical Association* 100 109–120.

Chen, Y., Dinwoodie, I., Dobra, A. & Huber, M. (2005b). Lattice points, contingency tables, and sampling. In *Integer points in polyhedra—geometry, number theory, algebra, optimization*, vol. 374 of *Contemp. Math.* Providence, RI: Amer. Math. Soc., 65–78.

Chen, Y., Dinwoodie, I. & Sullivant, S. (2006). Sequential importance sampling for multiway tables. *Ann. Statist.* 34 523–545.

De Loera, J. & Onn, S. (2005). Markov bases of three-way tables are arbitrarily complicated. *J. Symb. Comput.* 41 173–181.

Diaconis, P. & Efron, B. (1985). Testing for independence in a two-way table: New interpretations of the chi-square statistic (with discussion). *Ann. Statist.* 13 845–913.

Diaconis, P. & Sturmfels, B. (1998). Algebraic algorithms for sampling from conditional distributions. *Ann. Statist.* 26 363–397.

Dobra, A. & Fienberg, S. (2010). The generalized shuttle algorithm. In P. Gibilisco, E. Riccomagno, M. Rogantin & H. Wynn, eds., *Algebraic and geometric methods in statistics*. Cambridge University Press, 135–156.

Guo, S. & Thompson, E. (1992). Performing the exact test of hardy– weinberg proportion for multiple alleles. *Biometrics* 48 361–372.

Pak, I. (2000). On sampling integer points in polyhedra.

Shao, J. (1998). *Mathematical Statistics*. New York: Springer Verlag.